

OSHALIANG

Intellectual Property Law

Osha Liang LLP
Two Houston Center
Suite 3500
909 Fannin St.
Houston, TX 77010P 713.228.8600
F 713.228.8778
www.oshaliang.com
firm@oshaliang.com**FACSIMILE TRANSMITTAL SHEET****Date:** December 17, 2008**File:** 03226/417001**To:** Examiner M. Lee**Fax Number:** (571) 270-2648**From:** Aly Z. Dossa**Pages Including** 4
Cover:**RE:** Application No.: 10/824,450

☐ URGENT ☐ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ PLEASE RECYCLE

NOTES/COMMENTS:

Dear Examiner Lee:

Further to our conversation today, please find attached a claim set including the amendments we discussed. I authorize you to make the aforementioned amendments in an Examiner's Amendment. If you have any questions, please contact me at 713.890.1709.

Best regards,



Aly Z. Dossa (Reg. No. 63,372)

CONFIDENTIALITY NOTICE

This document (including any attachments) may contain privileged or confidential information. In the event that this document has been sent to you in error, or otherwise has been misdirected, please call the sender at 713.228.8600 to arrange for its prompt return or destruction. Your cooperation is greatly appreciated.

Houston • Paris • Silicon Valley • Tokyo • Austin

Application No.: 10/824,450

Docket No.: 03226/417001; SUN040872

EXAMINER'S AMENDMENT

1. (Currently Amended) A method for tracing an instrumented program on a processor having an x86 architecture, comprising:
 - triggering a probe in the instrumented program;
 - obtaining an original instruction associated with the probe, wherein obtaining the original instruction comprises searching a look-up table using a program counter value, wherein the look-up table comprises the original instruction associated with the probe and an address associated with the original instruction~~[[.]]~~, and wherein the program counter value corresponds to a current address of a first thread executing in the instrumented program;
 - allocating a second scratch space for a second thread;
 - loading the original instruction into the second scratch space, wherein the scratch space is allocated on a per-thread basis, and wherein a first scratch space for ~~[[a]]~~ the first thread executing in the instrumented program was previously allocated;
 - loading a jump instruction for the x86 architecture into the second scratch space wherein the jump instruction includes a next program counter value;
 - executing the original instruction in the second scratch space using ~~[[a]]~~ the second thread to collect data; and
 - executing the jump instruction in the second scratch space using the second thread.
2. (Currently Amended) The method of claim 1, further comprising:
 - emulating the original instruction to determine the program counter value if the original instruction is a control-flow instruction; and
 - returning control to the second thread at an address of the program counter value if the original instruction is ~~[[a]]~~ the control-flow instruction.
3. (Previously Presented) The method of claim 1, further comprising:
 - determining the next program counter value by incrementing the program counter value using a size of the original instruction.

Application No.: 10/824,450

Docket No.: 03226/417001; SUN040872

4. (Original) The method of claim 1, wherein the probe corresponds to a trap.
5. (Canceled)
6. (Canceled)
7. (Original) The method of claim 1, wherein the instrumented program is executed on a multi-thread architecture.
8. (Currently Amended) A system for tracing an instrumented program on a processor having an x86 architecture, comprising:
 - a first thread configured to execute the instrumented program;
 - a second thread configured to execute the instrumented program;
 - a first scratch space allocated for the first thread;
 - a program counter value corresponding to a current address of the first thread;
 - a look-up table arranged to store an address and a corresponding original instruction;
 - a trap handler configured to halt execution of the second thread when a trap instruction corresponding to a probe is encountered, use ~~an address of the trap instruction~~ the program counter value to obtain the corresponding original instruction from the look-up table, and ~~generate load~~ a jump instruction into a second scratch space to an address in the instrumented program, wherein the jump instruction includes a next program counter value;
 - ~~a first scratch space allocated for the first thread;~~
 - [[a]] the second scratch space arranged to store the original instruction and the jump instruction, wherein the second scratch space is allocated on a per-thread basis, and wherein the first scratch space for the first thread executing in the instrumented program was previously allocated; and
 - an execution facility for executing the original instruction in the second scratch space to collect data and executing the jump instruction, wherein the execution facility is a processor based on the x86 architecture.

Application No.: 10/824,450

Docket No.: 03226/417001; SUN040872

9. (Original) The system of claim 8, further comprising:
a buffer for storing the data.
10. (Currently Amended) The system of claim 8, further comprising:
a tracing framework configured to emulate the original instruction to determine a value of [[a]] the program counter if the original instruction is a control-flow instruction and to return control to a thread at an address of the program counter value if the original instruction is [[a]] the control-flow instruction.
11. (Original) The system to claim 8, wherein the trap handler sets a destination of the jump instruction to a next address immediately following an address of the trap instruction.
12. (Canceled)
13. (Original) The system of claim 8, wherein the instrumented program is executed on multi-thread architecture.